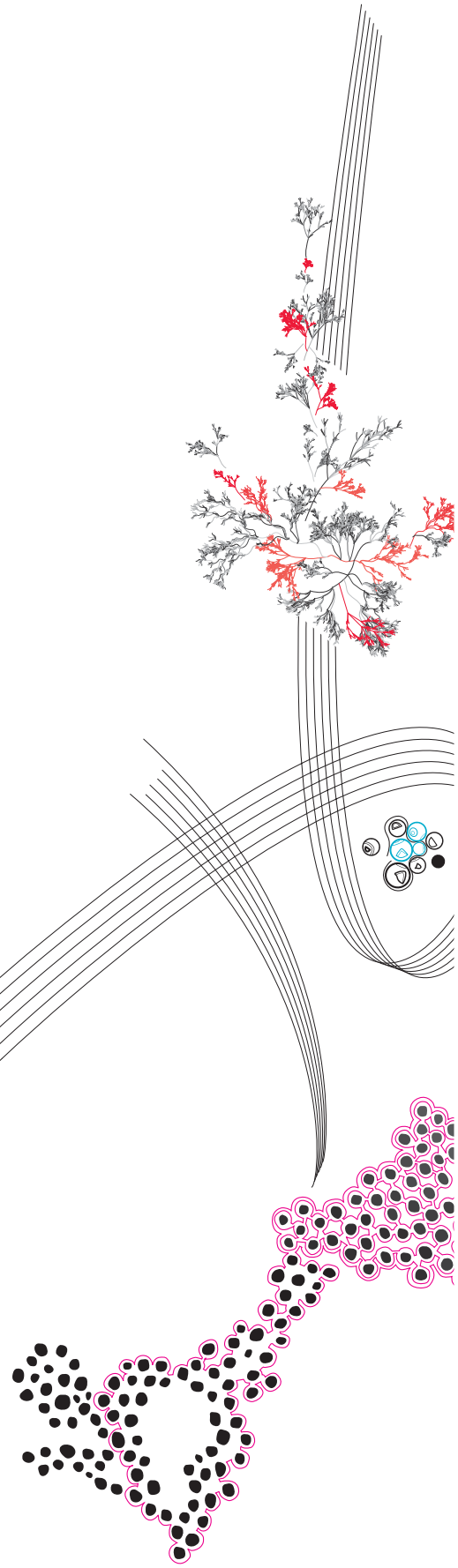


MSc Computer Science
Final Project

Safe Links or Safe Leaks? A Deep Dive into Information Exposure via Microsoft Safe Links on Public Sources

Resing, Max

Supervisor: Prof.Dr.Ir. Roland M. van Rijswijk-Deij
Dr.Ir. Mattijs Jonker
Dr.Ir. Raffaele Sommese
Dr.Ir. Thijs S. van Ede

November, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Chapter 1

Introduction

In contemplation of the results of this work, the graduation committee and I discussed the opportunity to submit the findings of the Master Thesis as a conference paper. Due to the problem space, the novelty of the work, the methodology and the temporal alignment with submission requirements, we decided to attempt a submission to the ACM Conference on Computer and Communications Security (ACM CCS) 2025. The submission deadline for the Call for Papers is in January 2025.

Although the Master Thesis is written in the form of a paper, the remainder of the document explains how my work and contribution meet the requirements for a Master Thesis set by the Examination Board. The conference paper without modifications of third parties is added as an appendix. With this deliverable, I intend to obtain a Master of Science in Computer Science with two specialisations in Cyber Security and Data Science.

Chapter 2

Requirements

The requirements for a Master's Thesis are listed below, followed by a statement on how I meet the requirements.

2.1 Scientific Quality

2.1.1 Interpret a possibly general project proposal and translate it to more concrete research questions

Originally, one of the supervisors suggested a thesis on investigating whether and to what extent Microsoft Safe Links make it into public mail archives. When exploring Safe Links further, I developed additional ideas for exploring the information leakage through Safe Links. The initial exploration led to the following research question:

- What are the implications when Microsoft Safe Links end up on public domain data sets?

Guided by the major research question, we developed four sub-questions:

- What kind of information does a Safe Link embed?
- What is the risk if a Safe Link exposes its embedded information publicly?
- How can we assess the scope of Safe Links escaping a controlled environment and ending up in the public domain?
- What are the implications of the exposed information if combined with contextual information of public data sets?

2.1.2 Find and study relevant literature, software and hardware tools, and critically assess their merits

While the topic is unexplored in academia, no directly related literature exists. We focussed on public documentation and discourse on the Safe Links product. Additionally, we briefly explored literature on information exposure through broader web technology that aligns with the issue observed with Safe Links.

2.1.3 Work in a systematic way and document your findings as you progress

During the work and the regular progress meetings, I kept track of "To-Dos" and filed them in an overview of open and past work. Additionally, I used a journal to document weekly progress and findings and take notes about the minutes of the meetings with my supervisors. After implementing and deploying the building blocks of the different aspects, I created pages to document the different research directions.

In the later stage, I often discussed progress on the paper and requested feedback for individual subsections.

2.1.4 Work in correspondence with the level of the elective courses you have followed

My Master of Science education covered the courses and requirements for two specialisations, with Cyber Security as the leading and primary specialisation and Data Science as the secondary specialisation. Since this work covers large-scale retrieval and organisation of privacy-sensitive data and several data processing, anonymisation and visualisation techniques, I argue that I sufficiently applied the learned theory of the Data Science courses. Moreover, this entire study fits into the domain of internet measurements and requires my learned understanding of network- and security-related protocols and systems. I showed how I applied theoretical knowledge from Cyber Security courses in a practical research. I used the regular progress meetings and the final delivery to demonstrate how I applied theory from the courses in practice.

2.1.5 Perform original work that has sufficient depth to be relevant to the research in the chair

My work presents a first-of-its-kind study on the exposure of information through Microsoft Safe Links on the public domain. During this work, I used several data science and internet measurement techniques relevant to the DACS group. My supervisors and I agreed that the insights are intriguing and concerning enough to even coordinate a responsible disclosure procedure with the National Cyber Security Center of the Netherlands. A responsible disclosure coordinating with multinational Cyber Security Incident Response Teams underlines the originality and depth of this work. Dr. Jeroen van der Ham-de Vos assisted in the responsible disclosure procedure.

2.2 Organisation, planning, collaboration

2.2.1 Work independently and goal-oriented under the guidance of a supervisor

My work was guided through a list of goals and milestones that I regularly revisited throughout this work. I prepared weekly progress meetings with an agenda and discussion points comprised of information from previous meetings and the newly achieved progress. We discussed achieved milestones and defined next steps. Often, my ideas shaped the next set of goals and milestones. I regularly updated the supervisors on achieved insights and we discussed next steps and deeper analysis efforts together.

2.2.2 Seek assistance within the research group or elsewhere, if required and beneficial for the project

The fact that I had three supervisors usually led to sufficient assistance and guidance. In one instance, I additionally reached out to external people to understand the purpose of an exotic value embedded in a Safe Link. It was a timestamp explicitly used in the .NET environment. This case demonstrates how I reached out to people beyond my supervisors to improve the study results.

2.2.3 Benefit from the guidance of your supervisor by scheduling regular meetings, provide the supervisor with progress reports and initiate topics that will be discussed

As mentioned, I prepared an agenda and discussion points before each meeting. This guided the meetings and I was able to benefit from detailed feedback provided through my supervisors. Meetings were usually weekly. A meeting comprised of a discussion on next administrative steps, and then we dived into the newly acquired results. At the end of a progress meeting, I had various pointers to work on.

2.2.4 Organise your work by making a project plan, executing it, adjusting it when necessary, handling unexpected developments and finishing within the allotted number of credits

Before starting the work, I had a project plan approved by my supervisors. However, the setup of some infrastructure introduced a delay. I underestimated the effort and discussed to readjust the schedule. Once recommitted to a new schedule, I experienced steady progress and learned that not every plan works out in its first version. In my opinion, the quality of the results benefited from the introduced delay nevertheless.

2.3 Communication

2.3.1 Write a Master thesis that motivates your work for a general audience, and communicates the work and its results in a clear, well-structured way to your peers

I added the final delivery as an appendix to this document. It contains the results of my work in the form of a conference paper. The paper is clear, concise, and structured in accordance with past papers submitted to previous iterations of the conference. The paper is well-structured, with a section on the introduction, background, methodology, and results. Furthermore, it provides a discussion and a related work section. The paper continues with a discussion on the ethics review performed by the University's ethics committee before presenting the conclusion.

2.3.2 Give a presentation with similar qualities to fellow students and members of the chair

I understand that a Master's Thesis concludes with presenting the work in a colloquium to fellow students and the graduation committee. I will present the work on November 26, 2024.

Appendix A

Paper

Below is the paper prior to any modifications from third parties.

Safe Links or Safe Leaks? A Deep Dive into Information Exposure via Microsoft Safe Links on Public Sources

Max Resing
University of Twente
Enschede, the Netherlands
m.resing-1@student.utwente.nl

Abstract

Microsoft rolled out a new security product called Safe Links in late 2014. The product leverages Microsoft's extensive threat intelligence to provide users with an automated and enhanced protection mechanism against phishing and malware distribution sites. Since the product primarily aims to protect customers of Microsoft products, it suggests that a Safe Link is constrained to a controlled environment where the data present in a link is already known to the user. However, the design of Safe Links makes the product prone to information exposure if it escapes this controlled environment and propagates to the public domain.

We present a first-of-its-kind study on Safe Links that escaped to the public domain. This work examines the various types of information encapsulated within a Safe Link and demonstrates how their exposure to the public domain compromises this sensitive data. Our findings reveal this issue dates back nearly a decade and impacts over 1,200 organizations globally. By combining the Safe Link encapsulated data with contextual information from their public sources, we substantially enhanced the scope of insights gained. To list examples, we concretely managed to associate corporate and private mail addresses and present how Safe Links enable Personally Identifiable Information (PII) exposure in governmental documents, potentially violating legal regulations.

1 Introduction

Headlines regularly dominate the news about how data from controlled environments was exposed to the public, often including sensitive information like login details, user accounts or protected PII. Often, information gets exposed through the malicious intent of threat actors. However, human negligence can also cause inadvertent information exposure, e.g., accidentally providing access to data or sharing data due to a lack of awareness.

Sometimes, inapt product design enables this inadvertent information disclosure. This work investigates the exposure potential through artefacts generated by the security product Safe Links. Microsoft developed the product to provide an automated solution to phishing and malware distribution. We demonstrate how the design decisions of this product make it prone to exposing its users' information.

Safe Links is a product that scans incoming emails for links. After identifying a link, Safe Link replaces the original destination. By replacing the destination Uniform Resource Locator (URL) with a wrapper link, any person who clicks the link will be first redirected over a Microsoft-controlled backend. The backend ensures time-of-click protection by inspecting a website before redirecting the user to the original destination [12]. The wrapper link encapsulates data, including PII, that is intended to be scoped between the user and Microsoft's mail services. The embedded information ranges from a timestamp, a tenant Identifier (ID), and a message ID to a recipient's mail address. These artefacts are known to a user and ensure the functionality of Microsoft products. But the moment a Safe Link escapes the controlled environment and becomes part of a public data set, the data embedded turns out to be sensitive, and information is exposed presumably involuntarily. Online resources confirm that this product has existed at least since 2015 [9].

In this paper, we study Safe Links that escape the limited scope between the user and Microsoft. We explain the implications of a Safe Link becoming part of the public domain and provide evidence that this is a common issue observed on the Internet. To back our arguments, we crawled selected public data sources and searched for Safe Links. We ended up with a repository of over 50,000 Safe Links extracted from public data. With these insights, we show substantial information exposure through Safe Links by combining the embedded data with contextual information of the originating data set.

As contributions of this paper, we demonstrated that:

(1) Over the past years, it became a widespread issue of Safe Links escaping their controlled environment and exposing information to the public.
(2) The majority of Safe Links is attributable to a few organizations
(3) Contextual information from the data set on which a Safe Link occurs reinforces the amount of information one can learn through Safe Links
(4) Encapsulated PII in a Safe Link present in documents disclosed through Freedom Of Information Acts (FOIAs) may potentially violate legal regulations.

Lastly, the results discussed in this paper lead to a responsible disclosure procedure in collaboration with the National Cyber Security Centre (NCSC) of the Netherlands. We disclosed our findings to Microsoft. Additionally, we contacted national Computer Security Incident Response Teams (CSIRTs) to adequately respond to the problem on national levels.

The remainder of the paper is structured as follows. We first introduce Safe Links in detail in Section 2. Section 3 explains how we crawled for Safe Links in public sources. The results in Section 4 elaborate on the demographics of the acquired Safe Links. We discuss Safe Link's directly embedded information and put it in context with the data set from which we sourced it. Section 5 discusses the sensitivity of the information exposure and elaborates on future work. We present related work and ethics in Section 6 and 7, before we conclude the paper in Section 8.

2 Background

Safe Links leverages Microsoft's extensive threat intelligence to strengthen protection against phishing and malware distribution sites. Figure 1 portrays the information flow of Safe Links. We distinguish between two events: First, there is the event of receiving a mail, i.e. ① to ④; then there is the process of clicking a Safe Link, i.e. ⑤ to ⑦.

On receiving a mail, a mail server scans the incoming mail for URLs (①). For each URL, the server instructs the Safe Links backend to wrap the original URL into a Safe Link (② & ③). The processed mail is moved to the inbox and the original URLs was replaced by the wrapper (④).

When a user reads the message and clicks the Safe Link (⑤), the client connects to the Safe Links backend first (⑥). The backend assesses the destination website on risks. When the website is confirmed safe, the user is forwarded, otherwise blocked or warned (⑦).

2.1 Composition of a Safe Link

For this work, it is essential to understand the embedded information of a Safe Link. A Safe Link can be a few hundred characters long and embeds information through percentage encoding defined in RFC 3986 [1]. The encoding makes the resulting link hard to read and hides the encapsulated data in noise. Figure 2 shows a fictional example of a Safe Link[1]. We highlighted relevant information.

Firstly, the subdomain defines the data centre region for a Safe Link. The URL is defined through a URL parameter. Next to the data centre region and the destination URL, we highlighted the recipient's mail address, the global message Universally Unique Identifier (UUID), the global tenant UUID, a timestamp and a 256-bit value at the end. We manually altered the different values of a Safe Link and discovered that it leads to the Hypertext Transfer Protocol (HTTP) response *400 - Bad Request*. Thus, we suspect that the 256 bits of Base64 encoded information in the *sdata* parameter functions as a checksum to validate a Safe Link. It is a standard output length for modern hash functions.

From the information present in a Safe Link, we can summarise the following risk of information exposure:

Firstly, a Safe Link embeds temporal information through the timestamp. Secondly, the original recipient of a link is embedded. The message identifier uniquely identifies a conversation stream. The tenant identifier precisely identifies the organisation which uses Safe Links. Lastly, the label of the data centre region provides a broad geographical and legal context in which an organisation operates.

2.2 The Controlled Environment of a Safe Link

Figure 1 shows that users interact with a Safe Link through their email inbox. The previous paragraphs also elaborate on information encapsulated by a Safe Link.

We understand that the information embedded in a Safe Link is information the user is allowed to obtain. To be concrete, a user is aware of the timestamp of receiving an email and knows the message's recipient, i.e., their own mail address. Also, a user is affiliated with the organisation to which the tenant ID belongs.

However, Figure 2 demonstrates how the data embedded in a Safe Link can be cluttered and disorganised to the human eye. Presumably, a user lacks awareness of

[1]We refrain from displaying a real Safe Link to prevent deanonymisation to conference reviewers

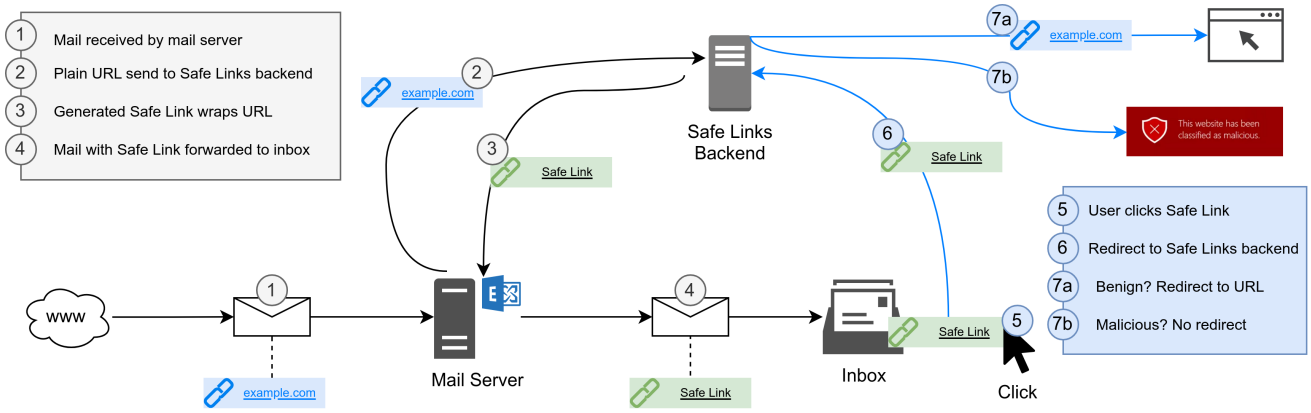


Figure 1: Illustration of how Safe Links replaces a URL on mail retrieval and what happens when a user clicks a Safe Link in their inbox. Colour encoding emphasises the distinction between the process of mail retrieval and the process of clicking a Safe Link.

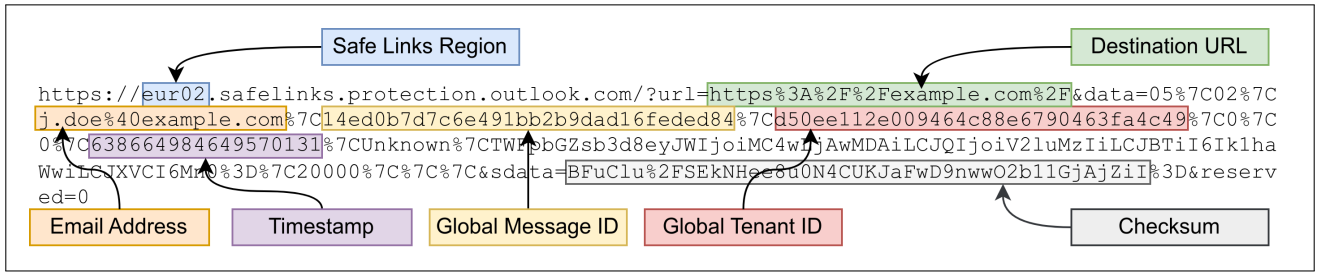


Figure 2: A fictional Safe Link pointing to the URL *https://example.com*. Visual assistance highlights relevant bits of information that we explore in this paper. Without it, locating the embedded information among the noise of characters is demanding.

the information in a Safe Link. Thus, they might involuntarily disclose sensitive information when they decide to share a Safe Link with third parties.

We learned a Safe Link...

- is primarily meant for a user's inbox.
- embeds information known to the owner of a mailbox but sensitive to 3rd parties.
- is opaque to a user on the embedded information.

Combining these facts suggests that a Safe Link was primarily designed to be scoped between the Microsoft platform and the user. We hypothesise that the product design did not sufficiently consider the possibility of a Safe Link leaving the user's control and becoming part of the public domain.

2.3 Absence of Mail Addresses

Although more Safe Links encapsulate a mail address, we also discovered many cases where a mail address is absent. We attempted to understand when a Safe Link encapsulates a mail address and when not, but we need help understanding it. Despite contacting Microsoft's customer service, we have yet to receive a meaningful response on how to prevent Safe Links from embedding a recipient's mail address. To the best of our knowledge, we are unsure whether the absence of a mail address is an intentional privacy feature. Due to our findings in Figure 11, we assume that embedding a recipient's mail address is not strictly a tenant-wide configuration but can be configured individually per mailbox.

3 Methodology

We systematically collect Safe Links from diverse data sources, including web content, mailing list archives and publicly accessible documents. While we collect a significant amount of data, our goal is not to exhaustively find every publicly available Safe Link. Instead, we collect enough data to assess the risks of Safe Links finding their way into the wild. To ensure we stay within the definition

of ethical research, we presented our methodology to the University's ethics committee. We discuss this more in Section 7.

3.1 Safe Links on Public Mailing Lists

We implemented a custom web crawler to identify Safe Links in public mailing list archives. Starting with a CommonCrawl index [14], we filtered on URLs matching common patterns of mailing lists, e.g. `/hyperkitty/` or `/pipermail/`. The crawler worked through the lists and downloaded archives younger than 2014. After downloading the archives, we extracted the desired information. The downloaded archives followed the standardized *mbox* format [16]. We scanned the archives from top to bottom, extracting the Safe Links and equally relevant information from the mail headers of the messages in which the individual Safe Links occurred. We included the sender's mail address, dates, subject lines, message identifiers, and the originating mailing list and mailing list host.

We retained the contextual information of mail headers to gain additional insights. Unlike the data embedded in a Safe Link, the mail header information is intentionally part of the public domain. We use the contextual insights of mail headers to amplify the amount of information we learn from Safe Links. The mail header provides a secondary data point to a dimension that the Safe Link exposes. Examples are (sender) mail addresses or timestamps.

3.2 Safe Links on Code Repositories

GitHub is an established collaboration platform for developers. We leveraged the GitHub API to crawl for common patterns present in Safe Links. By default, the API returns fewer than 1,000 hits for a query. Additionally, we have observed that different queries yielded overlapping search results. The limitations are intentional to prevent exhaustive querying of data from GitHub [5].

Due to these limitations, our results represent just a fragment of the actual scale of Safe Links on GitHub. We diversified the search results by building queries methodically. We combined the different Fully Qualified Domain Names (FQDNs) of data centre regions with search parameters, such as time periods or filters on the top 100 ranking GitHub organizations [7]. We intentionally left out searches of historical commit messages and code changes. We focussed on the current code files accessible on GitHub.

From the search results, we stored not only the source URL and the corresponding Safe Link. We also collected contextual information, i.e. the repository owner and popularity parameters like the fork and subscriber count of a repository.

3.3 Safe Links on Wikipedia

Wikipedia is an extensive, multilingual repository of user-contributed content supported by references to external sources. The Wikimedia Foundation – the non-profit organization that hosts Wikipedia – provides periodic snapshots of the databases. Each snapshot comprises the most recent revisions of all Wikipedia pages for a single language.

We selected the following languages: *Arabic, Chinese, English, French, German, Japanese, Portuguese, Russian,* and *Spanish.* The languages were selected based on speaker population and geographical diversity. We retrieved a snapshot for each language and scanned for Safe Links. The contextual enrichment of the Safe Links comprises the page title, the language, the revision number, a contributor's pseudonym, and the revision timestamp.

The data allowed us to examine Wikipedia pages with their revision history manually on Wikipedia. Reviewing the pages ensured that we grasped the exposure of information adequately.

3.4 Documents listed by Search Engines

We configured a metasearch engine to accumulate Safe Links indexed by various Search Engines (SEs). A metasearch engine uses third-party SEs from which it aggregates search results and presents them to the user. We leveraged it to investigate the exposure of Safe Links indexed by the three established SEs: Google, Bing and DuckDuckGo.

For each SE, we used the search operator `filetype:docx` and the FQDNs of the Safe Links regions. We limited the search to *docx* documents due to a limitation of our metasearch engine. We enriched each document with information on which search engine returned the result. Furthermore, we preserved the web server host and the original URL.

3.5 Data Sanitation and Additional Enrichment

Our repository contains tens of thousands of Safe Links, from which we filtered out malformed or duplicated entries. Furthermore, we used the Safe Links' global tenant identifier to query additional public information from Microsoft endpoints. With the 128-bit tenant UUID, we can query an organization's display name, a broad

Source	Metric	#
Repository	Safe Links	50.366
Safe Links	Exposed message identifier	13,766
Safe Links	Exposed mail addresses	3,102
Safe Links	Exposed tenant identifier	1.213

Table 1: Overview of distinct values being exposed through Safe Links based on a large-scale public crawl from selected public data sources.

regional context, and information about an organization's authentication setup. The data is accessible on two public endpoints[2].

4 Results

We have already mentioned how Safe Links are intended for the controlled environment between a user and Microsoft services. In this section, we reveal the widespread presence of Safe Links on public data sources. We also show that once in the public domain, a Safe Link exposes sensitive information like the user's email address. Furthermore, we demonstrate how the contextual information of a public data source in which a Safe Link is present reinforces the quantity someone can learn from the exposed information.

4.1 Demographics of Safe Links on Public Sources

Firstly, we want to demonstrate how Safe Link instances are frequent observations in the public domain. Albeit it was not our goal to exhaustively quantify the number of Safe Links in the public domain, we provide evidence of over 50,000 Safe Links on publicly accessible data. Table 1 provides key metrics of our repository and can be interpreted as follows:

- The message identifier represents an individual message in a mailbox.
- The mail addresses represent an individual recipient/mailbox.
- The tenant identifier represents an organization to which the Safe Link and exposed information are affiliated.

There is a stark contrast between the number of distinct Safe Links and the number of distinct mail addresses and tenant identifiers. Safe Links often originate from the same organizations and mailboxes.

[2]OAuth2 Information: https://login.microsoftonline.com/⟨tenant-uuid⟩/oauth2/authorize?client_id=⟨uuid⟩
OpenID Configuration: https://login.microsoftonline.com/⟨tenant-uuid⟩/.well-known/openid-configuration

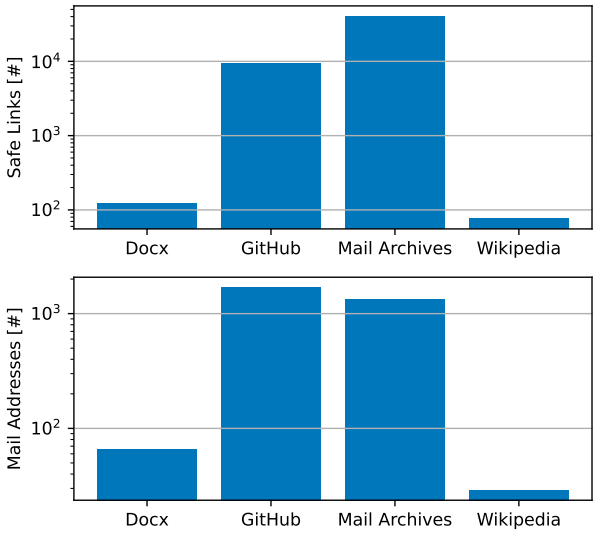


Figure 3: Overview of Safe Links and embedded mail addresses crawled from public data sources. The top image shows the number of Safe Links, and the bottom image shows the distinct embedded mail addresses.

While crawling for Safe Links, we also experienced that Safe Links are either more accessible or more likely to be present on some data sets than others. Figure 3 illustrates that our data acquisition returns much more Safe Links on mailing list archives and GitHub repositories than on public documents or Wikipedia pages.

Next, we look at when and from where Safe Links expose information. Figure 4 leverages the region label of the FQDN and the embedded timestamp. Our repository shows that the oldest Safe Link we scraped from the public domain is from the region *na01* and has a generation timestamp from late 2016. We also learn that almost all regions have a relatively recently generated Safe Link in the public domain. It shows that Safe Links finding their way into the public domain is a phenomenon that is not limited to certain geographical regions or specific groups of Microsoft customers.

The exposure of a mail address is likely the most striking type of information encapsulated in a Safe Link. We already discussed that not all Safe Links have email addresses embedded (compare Section 2.3). From Figure 5, we can learn that the data source has a noticeable impact on whether a Safe Link in the public domain exposes a mail address. Considering the whole repository, almost 65% of the Safe Links in our repository exposes the original recipient's mail address.

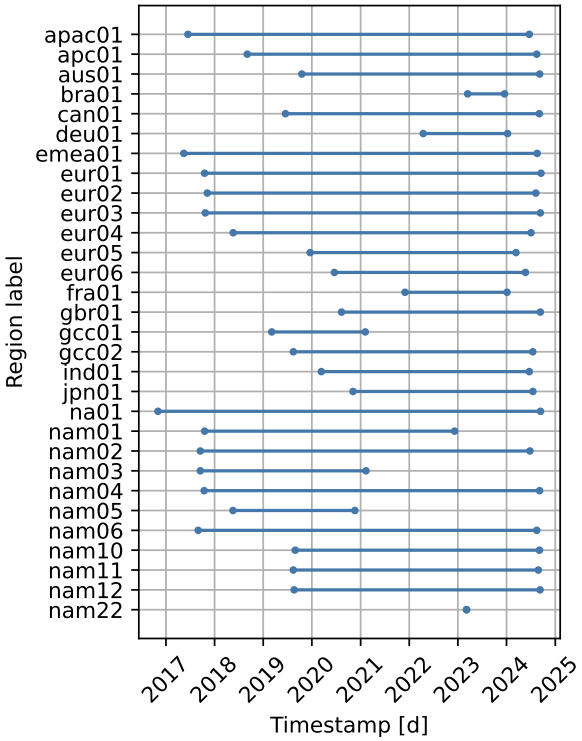


Figure 4: Evaluation of the timestamps embedded in Safe Links per individual data centre regions.

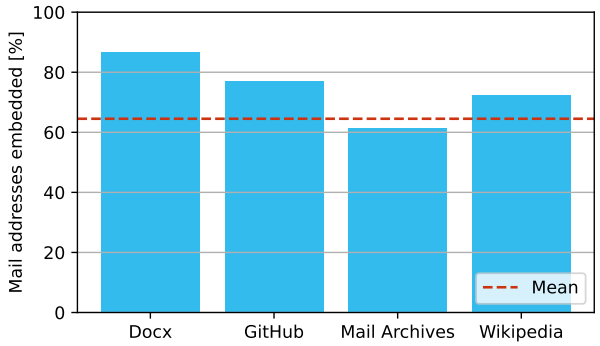


Figure 5: Percentage of Safe Links in which a recipient's email address is present

To conclude, Safe Links are regularly present on public data sets, although the number of Safe Links can vary greatly per data set. The encapsulated timestamp and region information reveals that information exposure through Safe Links is not limited to specific time frames or geographical regions. Safe Links – designed for a constrained environment between the user and Microsoft services – expose a user's mail address more often than not, i.e. around 65%.

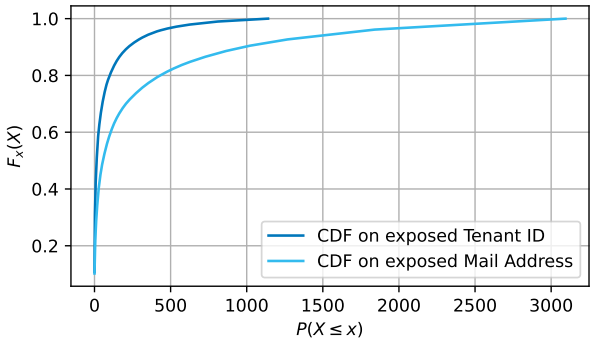


Figure 6: Consolidation of Safe Link instances on few mail addresses and mail hostnames. Both CDFs shows a long tail of mail addresses and hosts with a single circulating Safe Link

4.2 Concentration of Safe Links Exposure

We discovered a concentration of the Safe Link origins to comparably few organizations and mailboxes. We underline this argument by aggregating the number of unique Safe Links to the exposed information. Safe Links with the same exposed tenant ID originate from the same organization. Likewise, Safe Links with the same embedded mail address share the same originating mailbox.

Figure 6 plots the Cumulative Distribution Functions (CDFs) to display the consolidation of Safe Links on tenant identifiers and mail addresses. For example, just 15 organizations and 56 recipient mail addresses are responsible for 50% of the Safe Links we crawled from the public domain. The distribution shows a long tail with the remaining organizations and email addresses.

This section demonstrates a concentration of Safe Links in a comparatively small number of organisations and mailboxes. We learn about the organisations and individuals who repeatedly expose Safe Links to the public domain. Moreover, the section shows a long tail of single-exposing entities. It hints at how many organisations and individuals fail to identify the sensitive information in a Safe Link before sharing.

4.3 Safe Links on Mail Archives

In this section, we examine the exposure on mailing list archives. The mail headers provide context, which we combine with the encapsulated data of Safe Links. Our

Source	Metric	#
Mail Header	Message IDs	16,127
Mail Header	From Address	3,331
Safe Links	Unique Links	41,420
Safe Links	Exposed Message UUID	9,874
Safe Links	Exposed Mail Addresses	1,452
Safe Links	Exposed Tenant UUID	578

Table 2: Overview of data extracted from the crawl of public mailing list archives.

analysis provides the following critical insights: Firstly, we introduce a new metric Time-to-Exposure (TTE), which describes the time passed between the initial generation of a Safe Link and the moment it propagates to the public domain. The TTE also provides insights into how a dated Safe Link reappears in the public domain years after its initial generation. We also present that Safe Links, once part of the public domain, is archived online and, thus, persist for years. Lastly, the mail header information provided a second mail address, which entitled us to associate corporate and private mail accounts.

Table 2 describes the data set's composition. We discard messages which do not embed Safe Links.

Figure 7 shows a general tendency of growth in the number of messages with Safe Links embedded over time. We observe fewer exposed Microsoft message identifiers than mail header message IDs. It suggests that (potentially various) Safe Links from the same message in an inbox are shared multiple times on public mailing lists. The figure also reveals that Safe-Link-induced mail address exposure is closely coupled to the exposure of global tenant identifiers. Often, a new mail being exposed through Safe Links also exposes a new tenant identifier to the list.

Lastly, the figure incorporates the Safe Link exposed mail addresses ratio to the number of messages. The ratio lies stable at around 20-30%. We read that a new mail address is exposed in every fourth message with a Safe Link.

We understand that sensitive information and PII in the form of mail addresses are commonly observed on public mailing lists. The issue dates back years, and since online archives are long-living, the data remains accessible online.

4.3.1 A Safe Link's Time-to-Exposure. Next, we show how owners of mailboxes with the original instance of a Safe Link expose a Safe Link to the public domain much faster than Safe Links exposed by third parties. The TTE also helps us to show that sometimes a Safe Link surfaces on a public data set years after its initial generation.

We calculate the TTE as follows:

$$TTE = ts_{Mail\ Header} - ts_{Safe\ Link}$$

Moreover, with the header information, we can explore whether the sender's mail address matches the exposed mail address. We distinguish between the two cases:
Case A: The sender's mail address matches the exposed one. The case describes where the original owner of a Safe Link forwards it from their private mailbox to a public mailing list.
Case B: The sender address does not match the exposed mail address. The case describes when a Safe Link was shared privately and later exposed through a third party, i.e., not the Safe Link owner.

We compare the two cases in Figure 8. The X-axis shows the mail header timestamp. The Y-axis shows the TTE. We aid the observer with a highlighted region (coloured in teal). The area represents exposure events with a TTE between 1 to 10 minutes.

Clearly, *Case A*, where the sender address matches the exposed address, has a much higher concentration of Safe Links escaping to the public. The observation confirms that the initial owner of a Safe Link instance causes it to escape the controlled scope much quicker than someone who shares a Safe Link as a second or third instance.

The figure also reveals frequent exposure of a Safe Link to the public with 10^5 (ca. 69 days) or even 10^6 minutes (ca. 690 days) after the generation. The largest TTE observed translates to 830 days. We learn that a Safe Link can escape to the public domain even years after its initial generation. This observation shows the longevity of a Safe Link – and consequently the exposure of its embedded sensitive information.

Also, every point in Figure 8 represents an event of a Safe Link being exposed to the public domain. We observe again a growing concentration when we consider the mail header timstamp as the moment of exposure.

In this section, we have learned again that the number of Safe Links surfacing on public data sets is growing. Secondly, we have worked out insights on the posture of exposing Safe Links. We frequently observe that the original owner of a Safe Link propagates instances to a public mailing list minutes after retrieval. Lastly, we observed Safe Link instances on mailing lists where the mail header timestamp and the Safe Link timestamp are years apart. It suggests that occasionally, Safe Links are susceptible to escape to the public domain and consequently expose their information years after being generated.

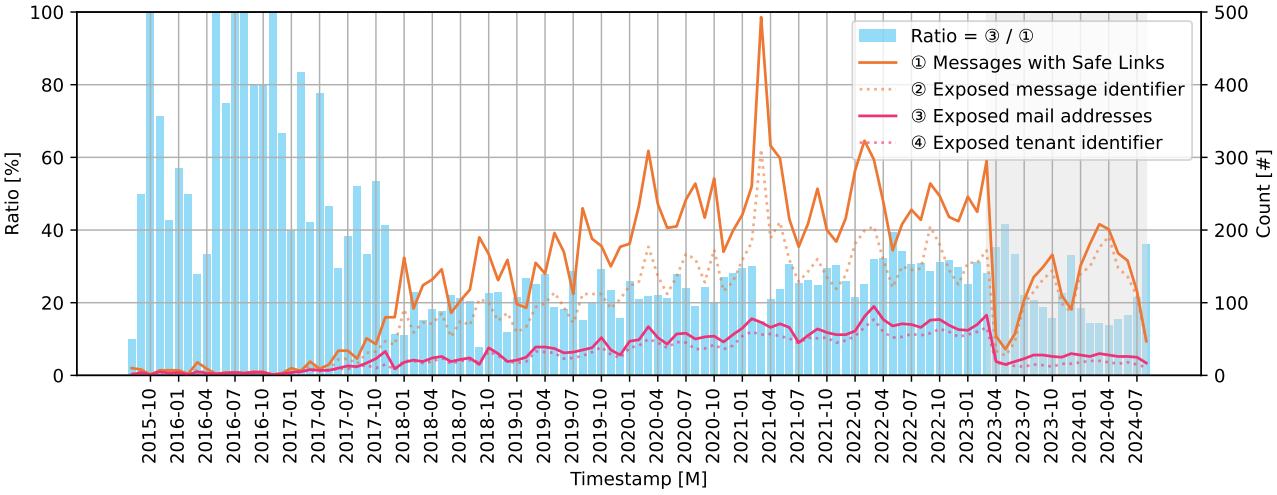


Figure 7: The plot compares the number of messages with Safe Links on mailing archives ① to the amount of exposed information. Displayed are the exposed number of Microsoft Global Message Identifiers ②, Microsoft Mail Addresses ③ and Microsoft Global Tenant Identifiers ④. We observed a crawling anomaly in the late stage of the crawl, for which we highlighted the affected period in grey. We suspect incomplete crawling during this period.

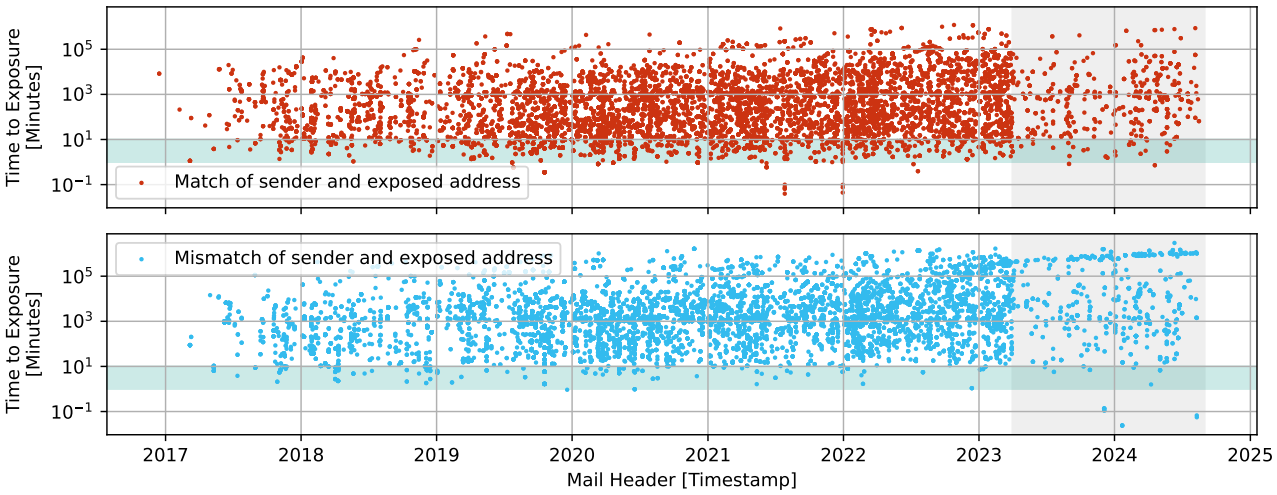


Figure 8: This plot compares the time interval between the timestamp of the email header and the Safe Link timestamp. The top plot displays cases where the sender address matches the Safe-Link-exposed mail address. The bottom plot displays the remaining cases.

4.3.2 Associate Corporate to Private Mail Addresses. The contextual information of a mail header also allows us to associate two mail addresses with each other. This information exposure is significant because it allows someone to attribute their professional email to a private email account.

Due to ethical requirements explained in Section 7, our repository masks the original username of a mail address through a hash, avoiding infringing on someone's

Source	Metric	#
GitHub	Repositories	1,446
GitHub	Organizations	1.174
Safe Links	Unique Links	9,436
Safe Links	Exposed Message UUID	3,819
Safe Links	Exposed Mail Addresses	1,750
Safe Links	Exposed Tenant UUID	777

Table 3: Overview of data extracted from the crawl of GitHub code repositories.

identity. We searched for exact matches on the usernames but different hostnames for the mail addresses. In our repository, we found 31 matching pairs. Examples are:

- *138[...]617@gmail.com* and *138[...]617@un.org*
- *137[...]805@gmail.com* and *137[...]805@jbtc.com*

We learn that not only do Safe Links escape the controlled environment of a corporate environment, but people also share links from corporate-received messages via private mail accounts and vice versa. Due to ethical concerns, our insights are limited to exact matches on hashes. It is safe to assume that human heuristics and plain-text usernames provide significantly more opportunities to associate a corporate and a private mail account with each other.

4.4 Exposure on GitHub

Safe Links are widely present on code collaboration platforms, which we underline with our almost 10,000 Safe Links fetched from public repositories. Table 3 provides an overview of exposed information on GitHub. Occasionally, multiple repositories hold Safe Links, which expose the same mail address. We counted the number of repositories on which a mail address appeared and found that the same mail addresses are occasionally present in several repositories. Figure 9 presents the results. Our repository's most common hostname of all mail addresses is *microsoft.com*, which exposes around 270 mail addresses on 250 repositories.

Collaborative coding duplicates a repository on various machines [2, 3, 6]. The number of forks indicates collaborators who cloned a repository, while watchers actively follow a repository. Our research discovered that code repositories with Safe Links are forked and watched hundreds of times, implying that a Safe Link is duplicated to and present on hundreds of personal devices.

We learn that Safe Links are very present on public code platforms. Collaborative coding practices duplicate Safe Links shared on popular repositories to hundreds of personal devices. It means the encapsulated data and PII are multiplied and redundantly stored on various

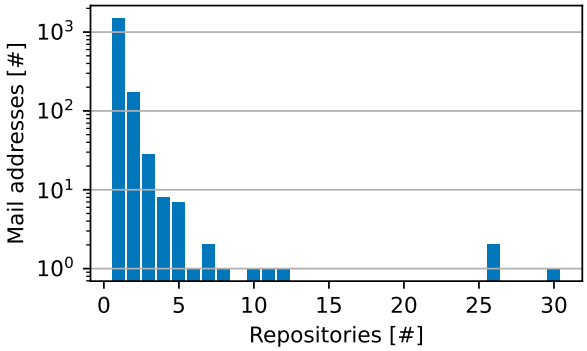


Figure 9: In our repository, most Safe-Link-exposed mail addresses occur in a single repository. However, occasionally, various Safe Links expose the same mail address in multiple repositories.

private devices – all due to the design of Safe Links and the fact that they are shared via code repositories.

4.5 Accessibility through Search Engines and Public Documents

Our results of this section will show how Safe Links are systematically indexed by SEs. Additionally, this section reveals that Safe Links present in FOIA disclosed documents raise potential legal concerns. While Safe Link enables the presence of email addresses in public documents, legal frameworks strictly regulate the issue of PII in FOIA disclosures [4, 13].

With the example of Bing, a popular SE, we demonstrate how SEs systematically indexes Safe Links and makes the information embedded in them easily accessible. The query *safelinks.protection.outlook.com* combined with a filetype search operator yields a concerning number of search results presented in table 4. We learn that Bing indexes Safe Links systematically.

Next, with the help of a metasearch engine, we acquired public *docx* documents through SEs. We instructed the metasearch engine to limit its search to Google, Bing and DuckDuckGo. Crawling results from all three returned an overlapping number of search results, implying that not only Bing but also other SEs widely index Safe Links.

Our crawl returned 88 distinct documents listed redundantly by multiple search engines. Various search results link to documents on national platforms handling FOIAs requests. We explored this further by visiting the

Document Type	PDF	Docx	Xlsx	Pptx
Results	10,300	102	32	9

Table 4: Overview of Bing Search Results as provided on October 26, 2024, when using *filetype* operator and the query *"safelinks.protection.outlook.com"*.

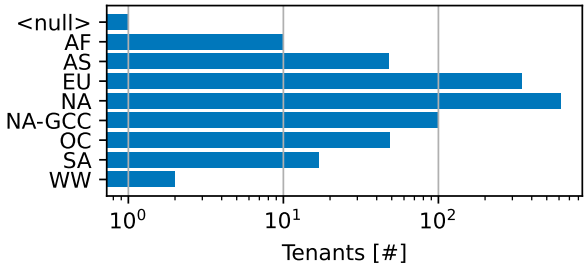


Figure 10: Tenant count per region. We leveraged the tenant identifier to query region labels from public endpoints.

websites of these platforms. Consequently, we found hundreds of documents with Safe Links using the platform-specific search features. The two screenshots in Figure 12a and 12b document our observation.

The Safe Links in our repository confirm that at least some expose a mail address – data considered PII. However, our limited insights of our data set do not allow us to quantify the full scale of the problem. Safe Links potentially violate legal requirements of FOIA disclosure since mail addresses pose as PII and FOIA regulations require the removal of PII prior to publication. The criticality of this finding motivated us to start a responsible disclosure procedure with the NCSC of the Netherlands.

This section highlights two key findings: Firstly, SEs provides easy access to Safe Links. Search engine operators enable someone to methodically refine their quest to locate Safe Links with concrete properties. Secondly, we identified hundreds of candidate Safe Links potentially exposing mail addresses in documents legally required to redact PII. The full scale is not yet known, but handled through a responsible disclosure.

4.6 Exposure on Wikipedia

We found that most Safe Links on Wikipedia pages are present in the English instance, followed by German pages. The manageable number of 76 Safe Links allows us to inspect the pages manually. We discovered that a page about a chemical compound was edited with a Safe Link that points to a corporate website. The embedded information in the Safe Link allows us to link the Wikipedia pseudonym to a human individual and shows how a Safe Link originating from a corporate inbox is embedded in a Wikipedia page pointing to the same corporate's website.

Another case shows how the same tenant identifier and mail address edit multiple pages of locations of the same constituency in Wales, UK. Lastly, we also found a single case of a vanity edit in which an exposed mail address matches the persona's name described on the Wikipedia page. Later, we found that newer revisions removed the Safe Link again.

4.7 Exposure of the Tenant Information

This section provides a deep dive into information exposed through the tenant identifier. We show a distribution of the tenants using Safe Links per region and we learn that the removal of mail addresses explained in Section 2.3 is not simply a global, binary configuration. Additionally, we leverage the tenant identifier to understand which institute uses the Safe Links product.

Figure 10 lists the number of tenants per region. Safe Links appear more commonly used in Europe, North America, and its governmental data centre subregion *GCC*[11]. Additionally, we learn the share of Safe Links that expose a mail address per tenant ID. While most organisations that use Safe Links have email addresses embedded either in all public Safe Links or in none, few organisations have a partial exposure of mail addresses. Perhaps the configuration to redact mail addresses from Safe Links can be configured per mailbox (compare Figure 11).

Using the tenant IDs and querying the tenant's name from open and public endpoints that Microsoft operates, we also learn about organisations using Safe Links. We tokenised the names and counted the occurrences. Most frequent tokens are *University, State, Inc., College, Group, Department, School, County, Services, Health.* Often, the tokens reveal governmental and academic institutions, but also legal entity types of companies.

We also learn information about a tenant's authentication features from the public endpoints. If Microsoft assumes that this information does not become public by simply being unable to guess the 128-bit tenant UUID, then Safe Links enables us to acquire this information methodically. We are unsure about the criticality of this discovery and made it part of the responsible disclosure procedure in which we communicated this with Microsoft.

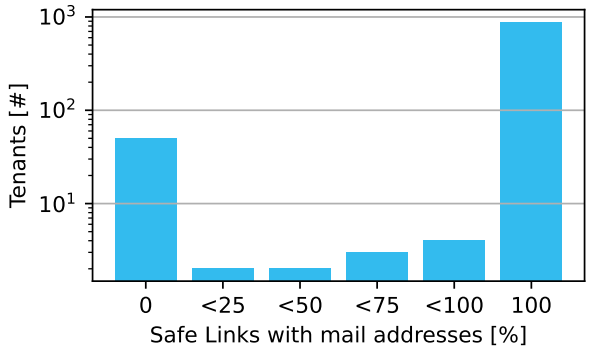


Figure 11: A histogram with the number of tenants, the associated Safe Links, and how many of those Safe Links embed a recipient's mail address.

To summarise this section, we have shown that the information in public Safe Links provides insights into organisations using Safe Links. We showed the geographic concentration of organisations and provided insights about the sectors to which these organisations belong. We disclosed the potentially sensitive information on a tenant's authentication features to Microsoft through a responsible disclosure process.

5 Discussions

We have sufficiently demonstrated that information encapsulated in a Safe Link becomes sensitive if it leaves the user-controlled environment and becomes public. In this section, we provide alternative design decisions and explain why this work might represent only a fraction of the true scale of Safe-Link-induced information exposure.

5.1 On the Design of Safe Links

Standard risk management suggests different strategies for dealing with risk, including risk elimination. It is unclear why Microsoft decided to embed all this information in a Safe Link, risking its exposure through events sufficiently presented in this work. Competing products like Proofpoint's URL Defense avoids embedding sensitive information in plain text. A trivial solution to reduce the risk of information exposure is eliminating non-essential information embeddings from a Safe Link. However, if removing such data would render a Safe Link inoperable, then it is at least advised to encrypt sensitive information. These suggestions are complicit with the proposed time-of-click protection Microsoft promises to offer with Safe Links[12].

Furthermore, the work often distinguishes between Safe Links with and without an embedded mail address. To the best of our knowledge, Microsoft's documentation does not explain how to avoid embedding a recipient's mail address. Also, Microsoft's customer support was unable to provide additional information. We recommend clearly documenting and communicating how to configure the product to prevent the embedding of mail addresses.

5.2 Estimate on the True Scale

The goal of this work was not to exhaustively crawl the public domain for Safe Links. We crawled from selected data sets to underscore our hypothesis of having sensitive information exposed when a Safe Link becomes public. To better estimate the actual scale, we searched many more platforms. Although we cannot quantify the issue completely, we found Safe Links on public platforms like Reddit, Mastodon, Shodan, Pastebin and a University's intranet. There is reason to believe that this work shows merely a fraction of Safe Links that found their way onto public platforms.

5.3 Responsible Disclosure

The insights acquired through this work require action from affected entities. We started a responsible disclosure procedure to disclose our findings. The responsible disclosure process is two-fold. We first brief Microsoft about the information exposure and the availability of potentially sensitive details about a tenant's authentication on a public endpoint. Later, we reach out to national CSIRTs to coordinate adequate responses to the presence of PII in FOIA documents on a national level.

5.4 Future Work

This work focuses primarily on the information embedded in a Safe Link. We also want to explore what information we can learn from the Safe Links backend. We know that the Safe Links backend interacts with DNS and web servers when a user clicks a Safe Link. We have already set up an infrastructure to measure these interactions. The study invites volunteers who use Microsoft products to interact with our endpoints through the use of Safe Links. Our infrastructure monitors the interactions of the Safe Links backend. So far, we presented our study design to the ethics committee of our university and received a green light. At the moment of writing the paper, we are in the process of recruiting participants for this study.

6 Related Work

To the best of our knowledge, there are no public studies on the information exposure through Safe Links. Due to the lack of related research, we briefly introduce public discussions about Safe Links.

6.1 Critical Voices

A user on the social media platform Reddit claims that the Safe Links design became a burden when sending out mails in bulk [15]. The user claims that multiple links to the same web server, and the number of subscribers to a mail newsletter amplify the number of requests on a webserver. Each receiving inbox that uses Microsoft Safe Links will trigger HTTP requests to a web server. The many requests resemble a HTTP application-layer Denial-of-Service attack.

Furthermore, Terence Eden blogged about some shortcomings in the product design [18]. He mainly criticises that the destination URL is unclear, and thus users revert back to copying the original Safe Link. The author explains the design decisions with a primary focus on acquiring telemetry and analytics. He is also concerned about the networking effects that come along with growing use of Safe Links. Safe Links position Microsoft as controlling proxy in between the user and the web.

Lastly, Ricardo Iramar dos Santos approached Microsoft as early as 2020 to address the information exposure of Microsoft Safe Links [17]. The company recognized the concerns as low risk and rejects them with a reference to the significant effort of exploiting certain information.

6.2 Information Leakage Through Web Technology

Safe Links is embedded in the broader domain of web technology for which studies on privacy implications exist. Krishnamurthy researched the information leakage through various web technologies [8]. The kind of leakage includes mail addresses, names, zip codes and even gender and health information. Leakage of this sensitive information occurred through various web technology, like cookies, parameters and HTTP referer links.

Another study conducted by Malandrino et al. assesses leakage of information from two perspectives [10]. They show how a tool can assist users in becoming aware of the exposure of PII. Secondly, they reverse engineered how data aggregators can and do build profiles based on privacy-sensitive information. As mentioned in the previous section, telemetry and analytics are one of the concerns mentioned.

7 Ethics

The sensitivity of the domain required us to submit an ethical review application to the research group's ethics committee. We presented our methodology and explained how we avoid infringing PII while processing the data. The use of one-way hash functions irreversibly anonymized data that potentially pose as PII. This methodology follows security best practices. In addition to approval of our methodology through the ethics committee, we also reached out to the Data Protection Office (DPO) of the University. With the DPO, we agreed on a sensible and sound method to archive our acquired data set.

8 Conclusion

In this paper, we performed a first-of-its-kind study of information exposure through Microsoft Safe Links. The product design suggests that a Safe Link should stay within a defined scope constrained to a user's mailbox, their organisation, and Microsoft services. We demonstrate how Safe Links are often not constrained to this environment and propagate to the public domain on a large scale. To underline the prevalent and far-reaching exposure, we crawled four selected publicly accessible data sources and built a repository of over 50,000 Safe Links scraped from the public domain. With this repository, we explored the directly exposed information in public Safe Links and introduced additional insights when combining embedded information with contextual data from the data sources. Concretely, Safe Links expose information in various regions for almost a decade. Furthermore, we found that code collaboration practices duplicate Safe Links and its embedded PII to thousands of personal devices. We could link corporate and private mail addresses and deanonymise Wikipedia pseudonyms. The repository also contained documents disclosed through FOIAs, in which Safe Links enabled the exposure of PII, potentially violating legal regulations. Lastly, the tenant ID helped us identify the organisations which have Safe Links enabled.

Finally, we questioned the design decisions, provided potential improvements, and gave pointers suggesting how our 50,000 Safe Links might represent a fraction of the actual scale of the problem. The results of this work were shared with stakeholders through a responsible disclosure procedure prior to publication. We coordinated the communication to Microsoft and national CSIRTs with and through the Dutch NCSC.

References

[1] Tim Berners-Lee, Roy T. Fielding, and Larry M Masinter. 2005. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986. https://doi.org/10.17487/RFC3986

[2] Marco Biazzini and Benoit Baudry. 2014. " May the fork be with you": novel metrics to analyze collaboration on GitHub. In *Proceedings of the 5th international workshop on emerging trends in software metrics*. 37–43.

[3] Scott Brisson, Ehsan Noei, and Kelly Lyons. 2020. We are family: analyzing communication in github software repositories and their forks. In *2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 59–69.

[4] Dutch Government. 2022. Wet open overheid.

[5] GitHub. 2022. Rate limits for the REST API.

[6] Jing Jiang, David Lo, Jiahuan He, Xin Xia, Pavneet Singh Kochhar, and Li Zhang. 2017. Why and how developers fork what from whom in GitHub. *Empirical Software Engineering* 22 (2017), 547–578.

[7] Takashi Kokubun. [n. d.]. Gitstar Ranking Organizations Ranking. Retrieved October 13, 2024 from https://gitstar-ranking.com/organizations.

[8] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. 2011. Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of the Web*, Vol. 2. 1–10.

[9] Sectigo Ltd. [n. d.]. crt.sh. Retrieved October 25, 2024, from https://crt.sh/?q=safelinks.protection.outlook.com.

[10] Delfina Malandrino, Andrea Petta, Vittorio Scarano, Luigi Serra, Raffaele Spinelli, and Balachander Krishnamurthy. 2013. Privacy awareness about information leakage: Who knows what about me?. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. 279–284.

[11] Microsoft. 2023. Office 365 Government. Retrieved November 17, 2024 from https://learn.microsoft.com/en-us/office365/servicedescriptions/office-365-platform-service-description/office-365-us-government/office-365-us-government.

[12] Microsoft. 2023. Safe Links in Microsoft Defender for Office 365. Retrieved March 25, 2025 from https://learn.microsoft.com/en-us/microsoft-365/security/office-365-security/safe-links-about?view=o365-worldwide.

[13] Office of the Australian Information Commissioner (OAIC). 2022. Freedom of Information Guidelines - Combined February 2022.

[14] Jay M. Patel. 2020. *Introduction to Common Crawl Datasets*. Apress, Berkeley, CA, 277–324. https://doi.org/10.1007/978-1-4842-6576-5_6

[15] Reddit User. 2022. Reddit: Microsoft Safe Links bombarding our website. Retrieved March 25, 2024 from https://www.reddit.com/r/sysadmin/comments/sphui0/microsoft_safe_links_bombarding_our_website/.

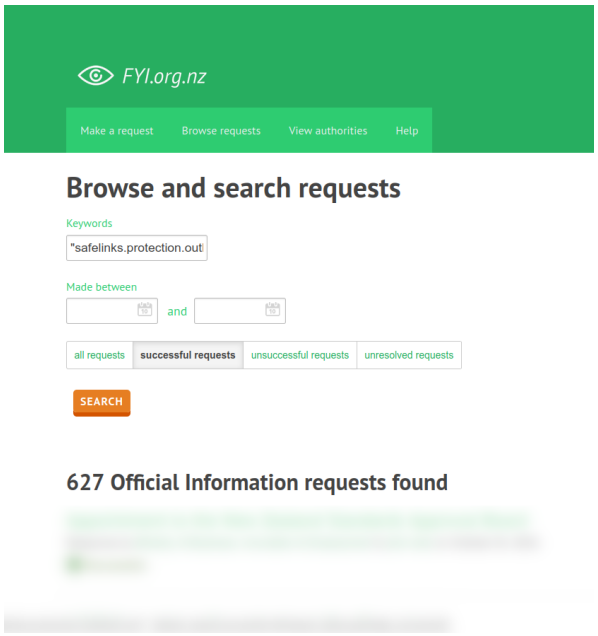
[16] Pete Resnick. 2001. Internet Message Format. RFC 2822. https://doi.org/10.17487/RFC2822

[17] Ricardo Iramar dos Santos. 2020. This is fine. Retrieved March 25, 2024, from https://ricardoiramar.medium.com/this-is-fine-6e032f497b8f.

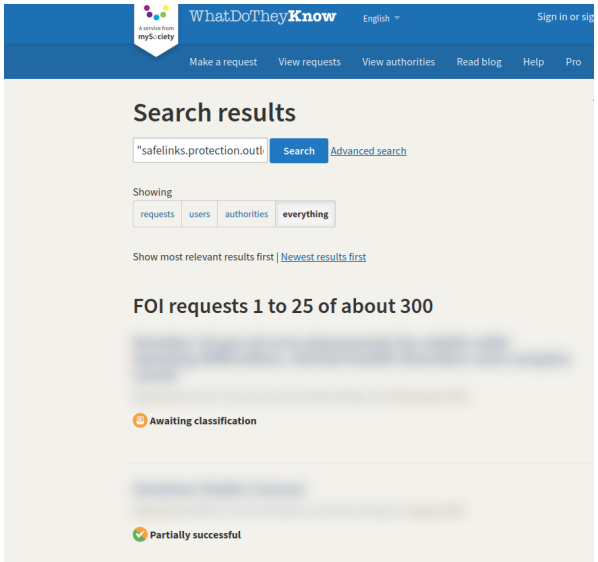
[18] Terence Eden. 2024. Safelinks are a fragile foundation for publishing. Retrieved March 25, 2024 from https://shkspr.mobi/blog/2024/02/safelinks-are-a-fragile-foundation-for-publishing/.

9 Appendix

9.1 Screenshots of FOIA Platforms



(a) Screenshot of an overview of search results on a FOIA platform for a nation in the global south.



(b) Screenshot of an overview of search results on a FOIA platform for a nation in Europe.

Figure 12: Two screenshots of search results on FOIA platforms when searching for Safe Links.